CHECKOUT & LAUNCH CONTROL SYSTEM

Application Software Process

Ben Bryant - Application Software Development

April 23, 1997

CLCS

Development Process

Define and implement functionality in increments to support defined capabilities (threads) :

- **Multiple CSCIs May Be Associated With Each Delivery Thread**

- **Development Phases Are Concurrent, Iterative and Overlapping:**
  - **"Capture" Functional Requirements From GOAL Software and Operational Experience (Transfer Functions That Work) and Define New Functionality Where Necessary**
  - **Implement Functionality Using COTS Tools/Languages to Fit Within the CLCS Architecture**
  - **Verify New Software Against Simulation Models (Via RTCN)**

- **Start of the Next Phase Is Not Dependent Upon Total Completion of the Previous Phase**

CLCS

Early Pathfinder Development

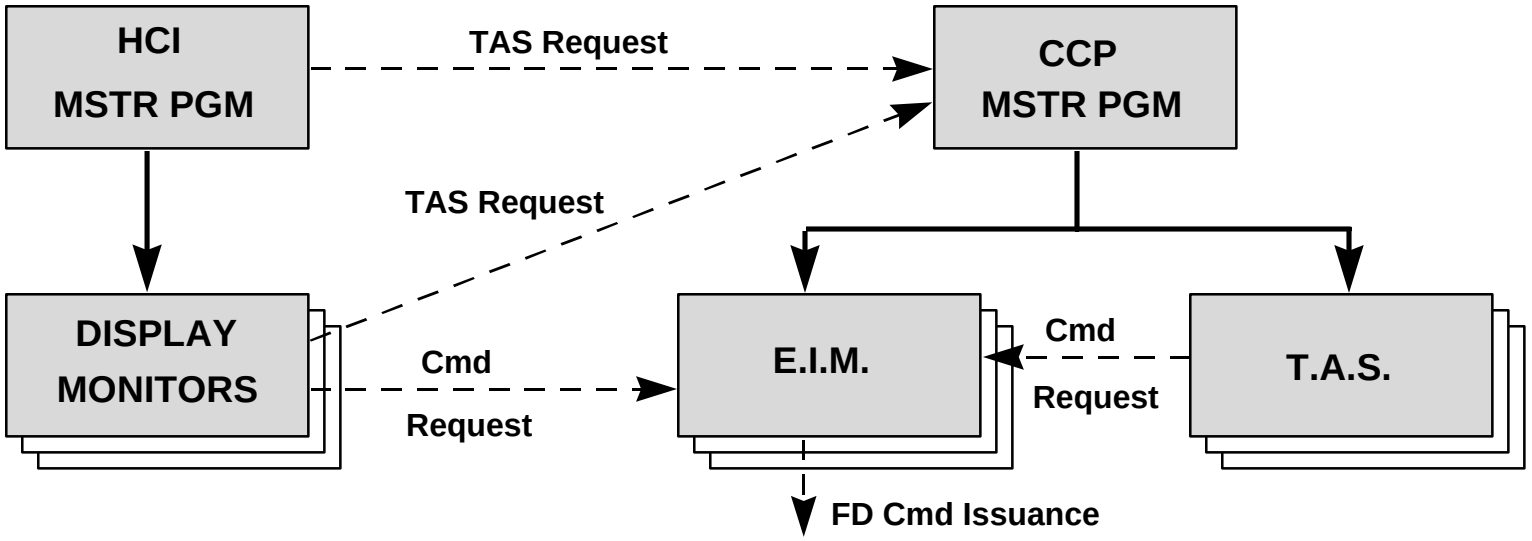
- **Pathfinder Teams Establish Foundation for Delivery Development:**
  - **Display Monitor Prototypes (Using SL-GMS)**
  - **Common Display Object Library**
  - **Generic Application Software Set Architecture**
  - **Programming Standards**
- **Super Lightweight Tank Development (in Progress)**
  - **Validates Display Monitor Concepts, Process and Standards**
  - **Support Tanking Tests (Monitor Only)**
- **OPF Power-Up/Power-Down -- Integrated Systems Pathfinder**

Early Pathfinder Development

- **HMF OMS/RCS - Initial Pathfinder (in progress)**
  - **Validates Application Software Set Architecture and Inter-Process Communication Protocols**
  - **Validates Development Approach and Process**
  - **Validates COTS Tools/Languages Selections (Control Shell, Paradigm Plus, Java)**
  - **Development (in Progress)**
    - **Pathfinder Content Will Go Through the Design Panel Process**
    - **FRCS Requirements Complete. OMS Requirements Will Follow After Concept Validation**
    - **Design Work in Progress on Architecture Common Components**
    - **Design Work in Progress on Mapping HMF Requirements to Architectural Elements**

CLCS

Application Software Set Architecture



- **Diagram Only Shows Major Communication Paths**
- **Details/Functions of Each Architecture Element Are Still in Development**

Development Environment

- **Display Monitors**
  - **SL-GMS to Be Primary Tool for Display Monitor Development**
  - **Alternatives Being Investigated to Support Displays That Do Not Require the Power of SL-GMS**
  - **Development Will Be Done on PC's, HP-UX, and SGI O2 Workstations**

- **End Item Managers**
  - **Paradigm Plus to Be Used for Requirements Capture and Design Phases**
  - **RTI's Controlshell and Java Are Being Evaluated As Implementation Tools**
  - **Expect Experience Gained Over 3 Month HMF Pathfinder Effort to Aid in the Selection of One Implementation Tool**
  - **Concern Over Safety and Complexity of C++ Drove Consideration of Java As a Risk Reduction Path**
  - **Development Will Be Done on SGI O2 Workstations**

- **Test Application Scripts**
  - **Development Is TBD at This Time**

CLCS

Product Vs. Delivery Matrix

CSCI	Redstone	Thor	Atlas	Titan	Scout	Delta	Saturn	Nova	RLV
SLWT	X								
Domain Libraries	X		X	X	X	X	X	X	X
HMF OMS/RCS	X	X	X	X					
EPDC				X	X	X			
ECLSS				X	X	X			
DPS				X	X				
ISL				X	X				
INTG				X	X	X			
APU/HYD				X	X	X	X		
COMM/NAV				X	X	X			
DPSME				X	X	X			
GNC				X	X	X	X		
MEQ				X	X	X			
MPS/SSME				X	X	X	X		
OMS/RCS				X	X		X		
FSW				X	X	X			
PLDTST				X	X	X			
PRSD/FC				X	X	X	X		
SRSS				X	X	X	X		
HPU/HYD							X		
ECS/PVD				X	X	X			
LACC						X			
HGDS/HWS						X			
LH2							X		
LO2							X		
GLS								X	
CCS									X

This matrix maps today's ASWTs (or CSCIs) to deliveries where they will provide capabilities. NOTE: ASWT structure is under review by Shuttle Engineering which will affect CLCS CSCI definitions.

Data Analysis and Presentation

Real-time Advisory

- **Integrate Existing Advisory Systems into CLCS**
  - **OPUS**
  - **APU Neural Network**
  - **APU, HPU, MPS High Speed Display**
  - **PAT**
  - **JGOAL**
  - **Others Under Considerations**
    - **KATE**
    - **PSA**
    - **DLES**
- **Interfaces With COTS Packages**
  - **SL-GMS**

CLCS

Data Analysis and Presentation

Support Advisory

- **Robust Web Interface to existing CAP Programs**
  - **Java Based**
  - **Begin With Caps 104,145,134,135 for Redstone**
  - **Proceed With Other Caps for Later CLCS Releases**
- **Advanced Data Analysis Tool**
  - **Seamless Java Based Solution to Users Data Analysis and Presentation Needs in Both the OCR and the Office.**
  - **Advanced Features (I.E. Zoom, FFT, Linear Regression, History -Vs- Real Time, Online Stripchart, et al)**
- **Interfaces with COTS packages**
  - **MS Excel**
  - **SL-GMS**
  - **LabView**

CLCS